P170M109 Computational Intelligence and Decision Making

Unsupervised Learning

ktu 1922

2020

Machine Learning Approaches

- **Supervised learning** – learning from input-output pairs to map from input to output
- **Unsupervised learning** - learning patterns without explicit feedback
- **Reinforcement learning** – learning from a series of reinforcements (rewards or punishments)
- **Semi-supervised learning** – learning from few labeled examples and a large collection of unlabeled examples
- …

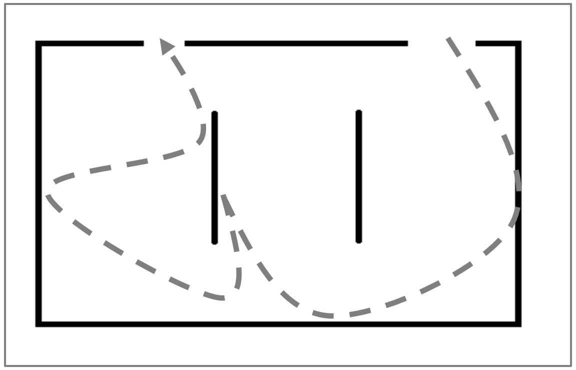
Machine Learning Approaches

	Supervised Learning	Unsupervised Learning	Reinforcement Learning
Dataset	fixed; correct (expected answer is given)	fixed; no label or class information given; no explicit feedback	incremental; feedback (reward) about the value of the action in environment
Objective	predict answers for new examples	extracting information	find a suite of actions to maximize its accumulated reward
Applications	regression (prediction); classification; translation	feature extraction; data compression; series modeling; clustering; anomaly detection	game playing; robot navigation

Advantages of Unsupervised Learning

- **No labels needed** – annotating (manual labeling) large datasets is expensive;
- **No prior knowledge is needed** about how many or what classes the data should be divided into.
- **Enables to gain insight into the structure of data** (correlations and relationships of attributes) before designing a model in supervised learning.

Example



Customer segmentation.

Given: data on the buying patters of all customers of a grocery store.

Objective: suggest special offers or plan store layout.

Unsupervised learning is applied to identify buying patters.

For example, those who:

- buy coloring pencils, also buy children's books
- buy frozen food, also buy snacks

Clustering

- Partitioning the entire dataset based on some similarity criterion (distance)
- Finding unknown groups of elements so that data within a group (or else, cluster) are similar to each other and dissimilar to data from other clusters

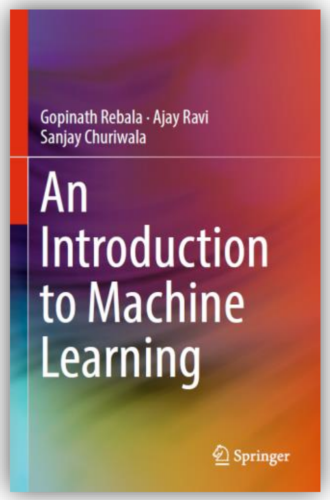
Applied in:
- Detecting groups of data across multiple attributes
- Data reduction tasks (data compression, noise smoothing, outlier detection, dataset partition).

Classes of Clustering Methods

- **Hierarchical methods**
  - <u>Agglomerative clustering</u> *(start with a singleton clusters and merge with respect to the similarity criterion)*
  - <u>Divisive clustering</u> *(start with all elements in one cluster and divide in several partitions in each step)*
- **K-means methods**
- **Graph theory methods**

Literature

Rebala, Gopinath, Ajay Ravi, and Sanjay Churiwala. *An Introduction to Machine Learning*. Springer, 2019.



…
Chapter 6, Clustering
…

https://link.springer.com/book/10.1007%2F978-3-030-15729-6

Use KTU VPN or perform search through https://vb.ktu.edu (uses SSO login and proxy to access full text document)

K-Means Clustering

Given:

S – set of elements

K – Number of clusters

Method:

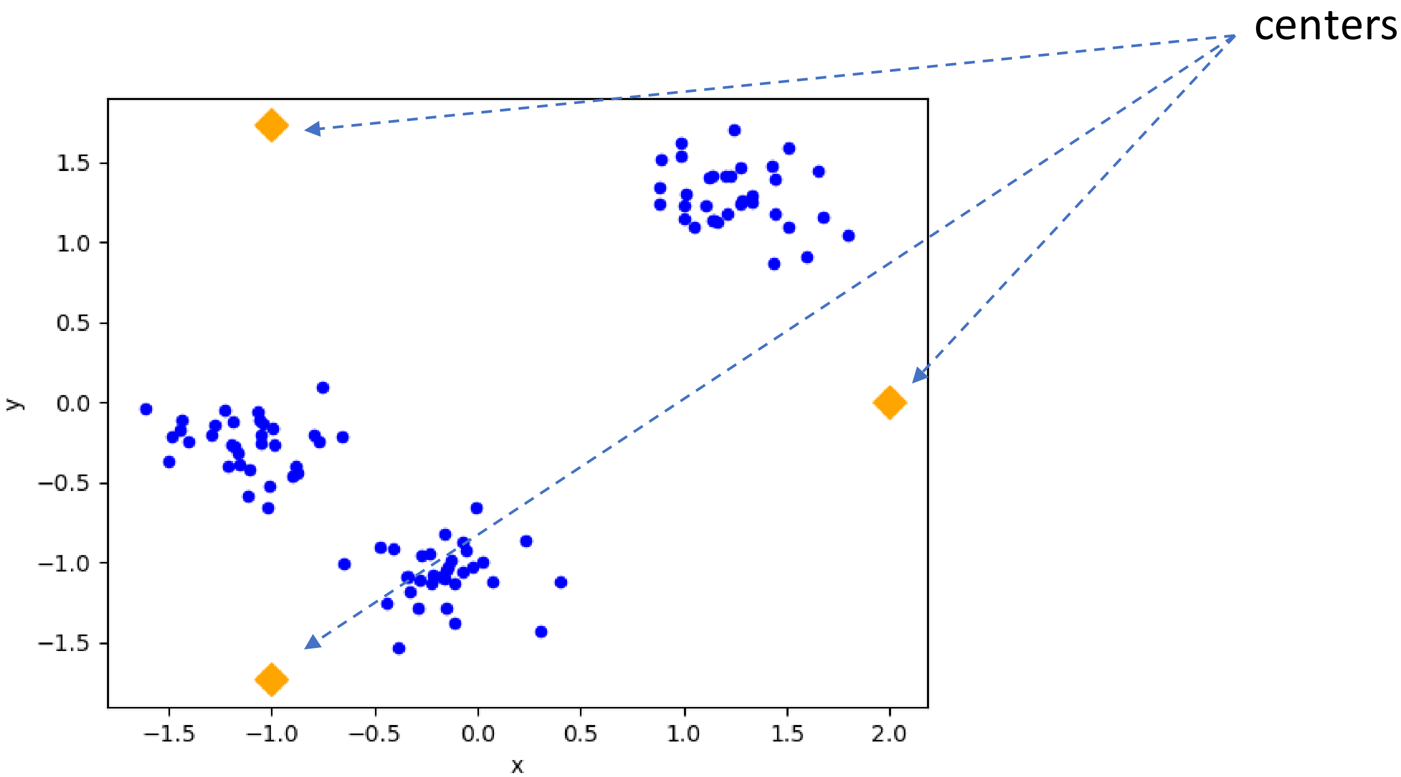
Step 1: Select K initial centers of clusters $C_1, C_2, \ldots, C_K$.

Step 2: Assign each element $X \in S$ to the cluster C_i $(1 \leq i \leq K)$ with the closest center.

Step 3: Recalculate the centroids in each cluster C_j $(1 \leq j \leq K)$ in which element was added or removed.

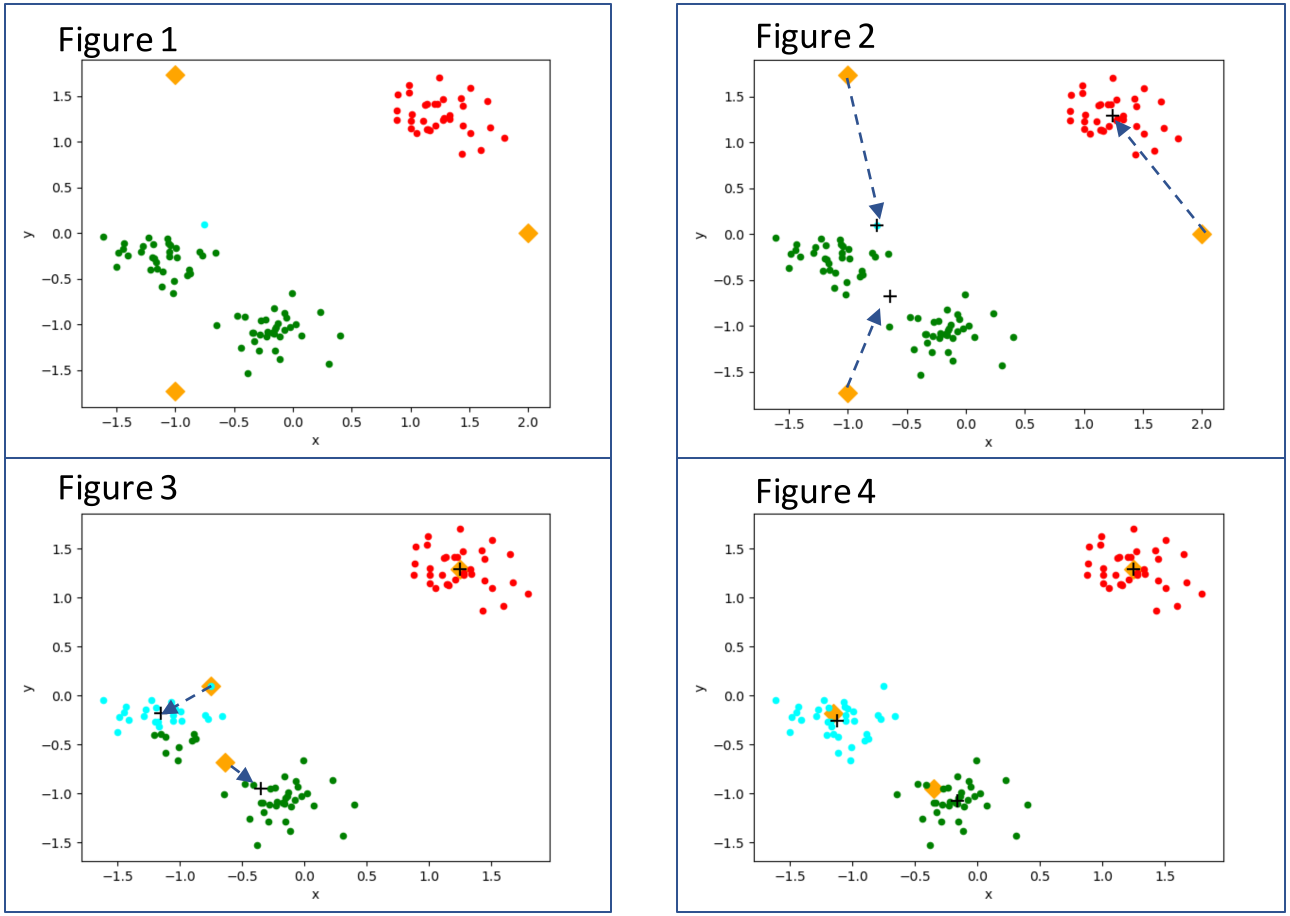
Step 4: Repeat the steps 2 and 3 until the algorithm converges.

K-Means Clustering



Number of clusters = 3

K-Means Clustering



current centers

+ updated centers

K-Means Clustering

How to…

- evaluate similarity (distance, cost function)?
- choose number of clusters?
- initialize centroids?

K-Means Clustering. How to evaluate similarity?

Euclidean distance:

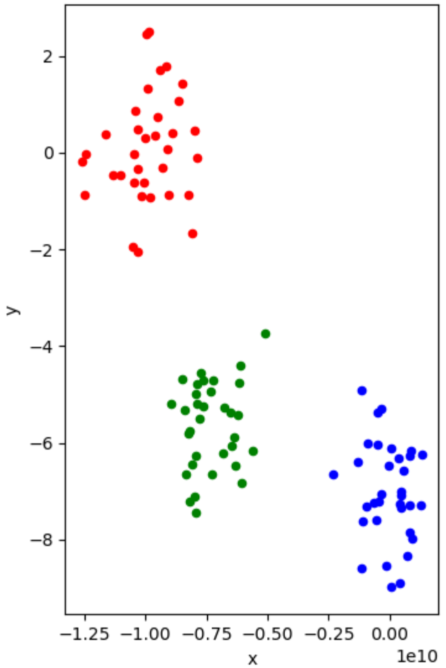
$$d_j = \sqrt{\sum_{i=1} (X_i - C_i^j)^2}$$

d_j - distance of element X to the center C^j

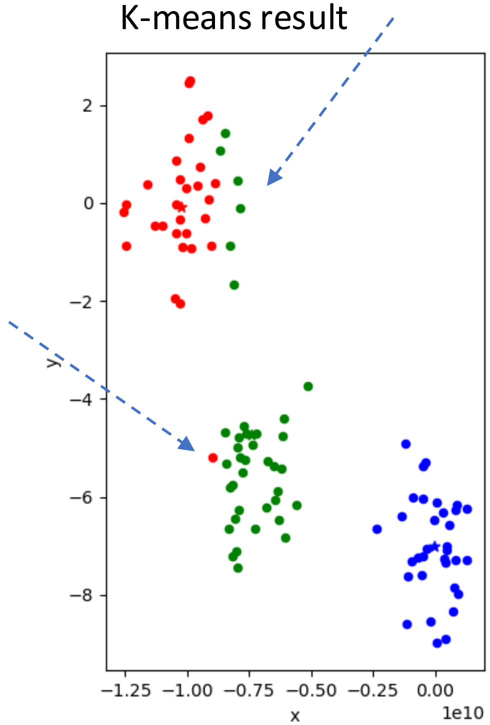
K-Means Clustering.

Standardization (or normalization) is important!

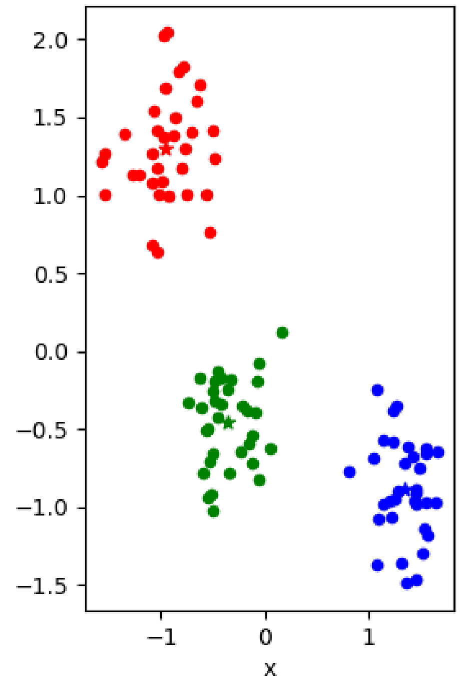
Simulated clusters



K-means result



K-means result for standardized dataset



K-Means Clustering. How to choose number of clusters?

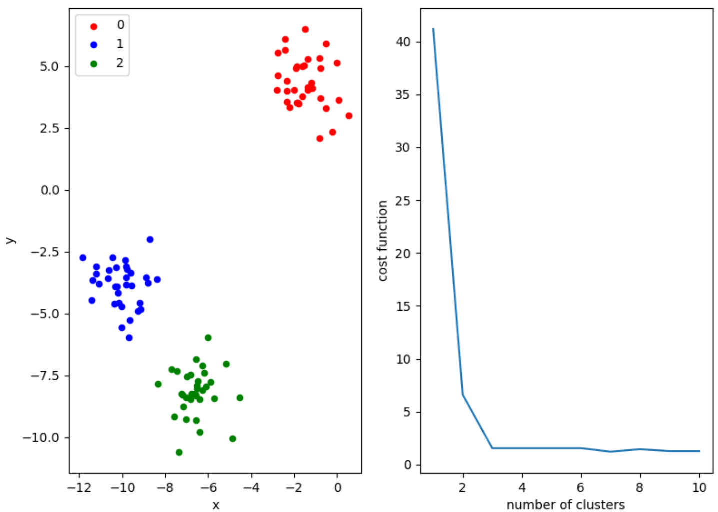
- Elbow method
- Silhouette method
- Gap statistic method
- …

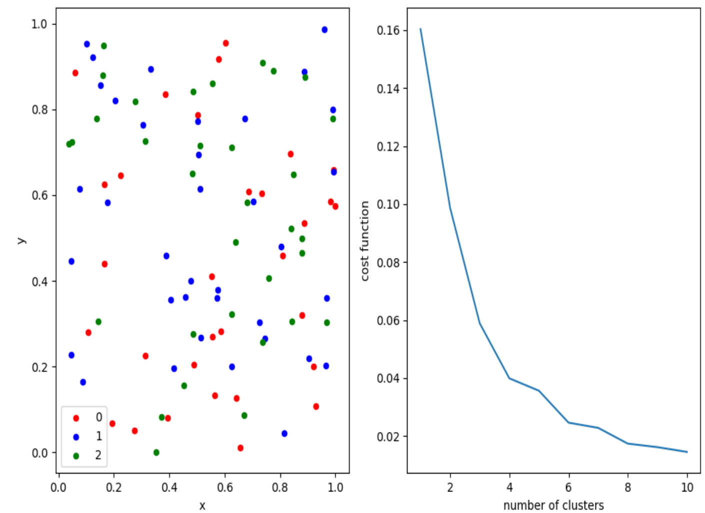
K-Means Clustering. How to choose number of clusters?

Elbow method:

Inexact!

- plot cost function vs number of clusters;
- look for a change of slope from steep to shallow.



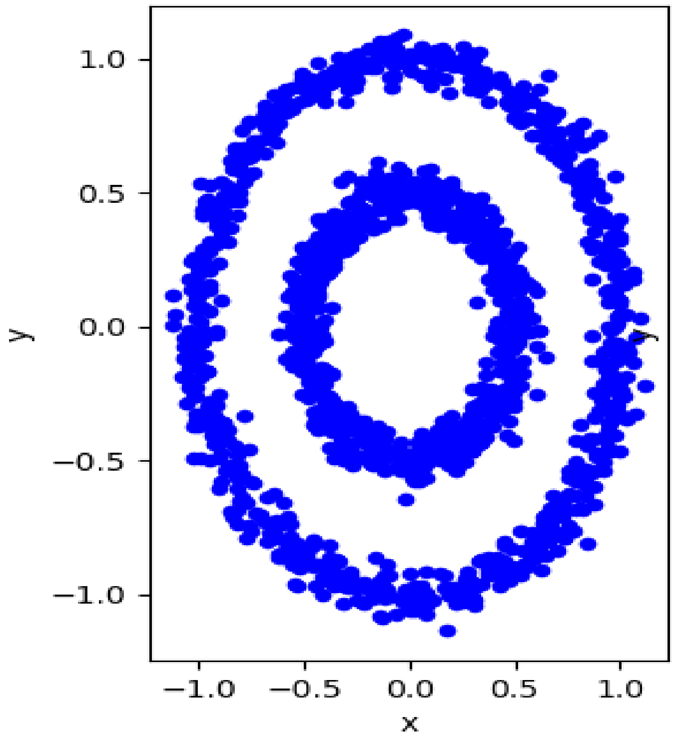


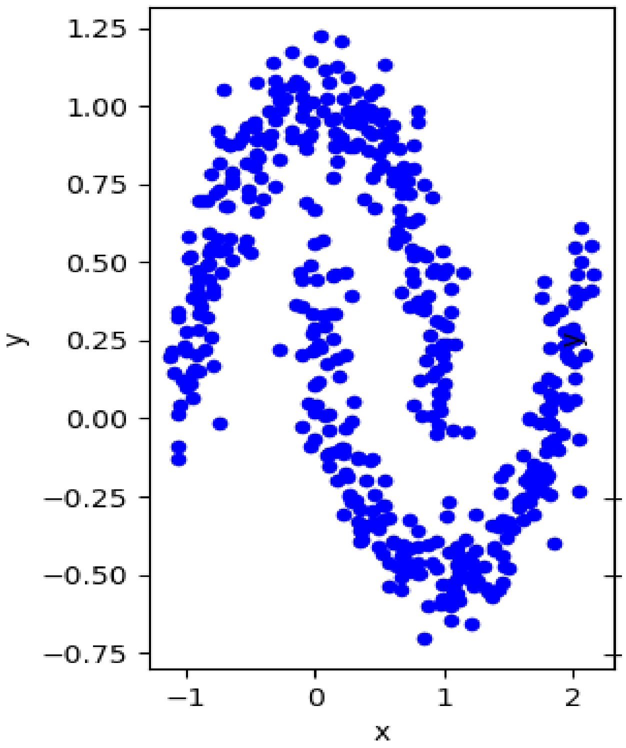
K-Means Clustering. How to initialize centroids?

Suggested solutions:

- Randomly partition data into K non-empty clusters and calculate centroids.
- Randomly select K elements from the dataset to represent clusters.
- Select K elements from the dataset to represent clusters. The elements should be as far as possible from each other – difficult to implement in high dimensions.

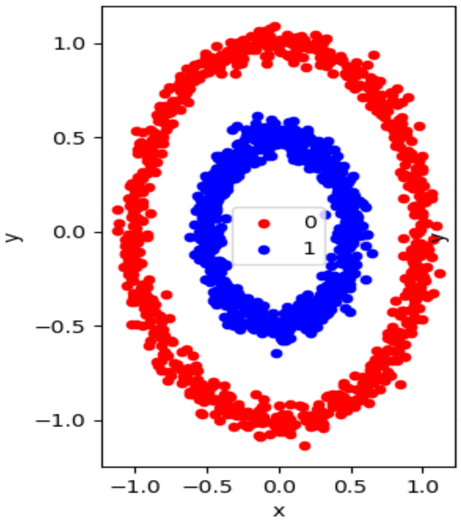
K-Means Clustering. Is it suitable for all datasets?



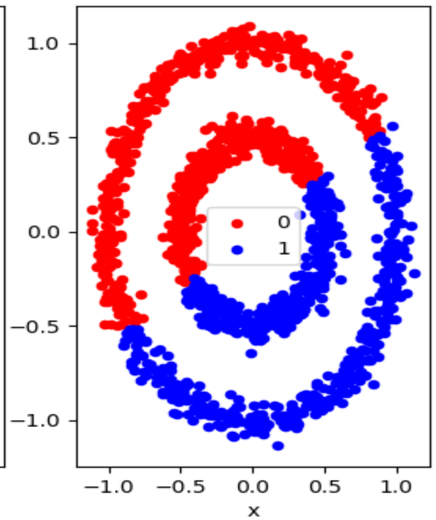


K-Means Clustering. Is it suitable for all datasets?

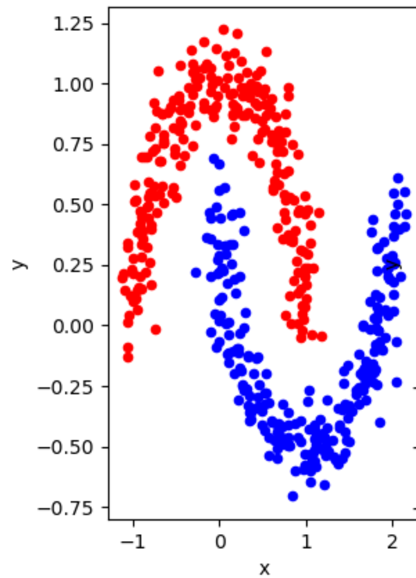
Simulated clusters



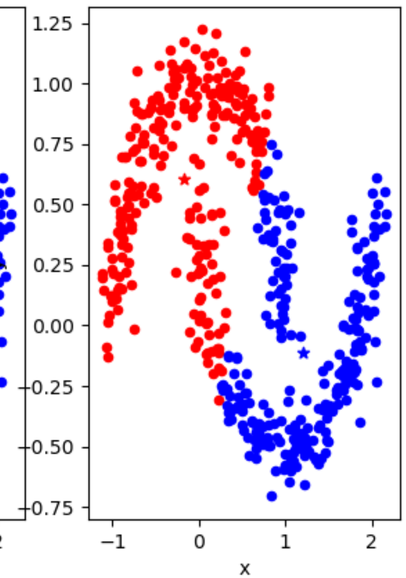
K-means result



Simulated clusters



K-means result



K-Means Clustering

Strengths:

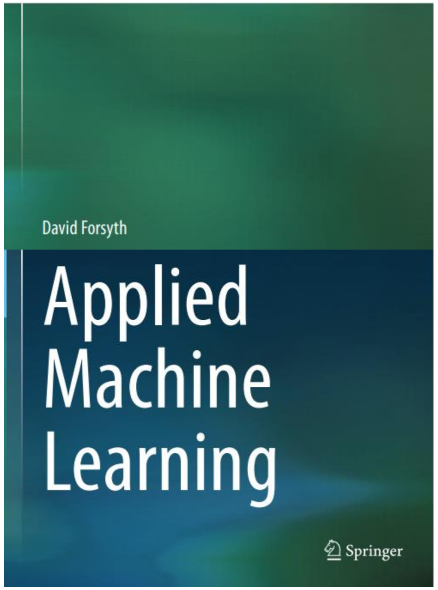
- simple implementation.

Weaknesses:

- applicable only to data objects in a continuous space.
- the number of clusters K must be specified in advance.
- not suitable to discover clusters with non-convex shapes as it can only find hyper-spherical clusters.
- sensitive to outliers.

Literature

David Forsyth. ***Applied Machine Learning*****. Springer, 2019.**



…
8.2.2 Clustering -> Soft Assignment
…

https://www.springer.com/gp/book/9783030181130

Soft Assignment (K-Means Clustering)

Let's say, You are clustering images based on it's extracted numerical features, and the results are:

What difficulties may occur during data analysis?

Soft Assignment (K-Means Clustering)

Using standard k-means clustering each point (feature vector) must belong to exactly one cluster!

Soft Assignment (K-Means Clustering)

Let's assign point $\mathbf{x}_i = \{x_1, \dots, x_n\}$ to cluster centers with weights $\omega_{i,j}$. Here

- $\mathbf{x}_i$ –point represented features vector, where $i \in [1, N]$, here N – number of points
- $\omega_{i,j}$ - connects point i with cluster center j.
- $j \in [1, c]$, where c is total number of clusters
- all weights are non-negative: $\omega_{i,j} \geq 0$
- each point should carry a total weight of 1. It means: $\sum_j \omega_{i,j} = 1$

i.e. we have image represented by it's features as $\mathbf{x}_i$ vector. Here one part of image is urban area, another part is forest. One of the scenarios after clustering that image would have similar weights in two clusters, it means:
$\omega_i = \{0, \dots, \omega_k, \dots, \omega_l, \dots 0\}$ – here k, l is the clusters represented forest and urban area (in this case $\omega_k \approx \omega_l \approx 0.5$).

Soft Assignment (K-Means Clustering)

Objective to minimize: $$\Phi(\omega, \mathbf{c}) = \sum_{i,j} \omega_{i,j} \left[(x_i - c_j)^T (x_i - c_j) \right]$$

Relation between $\boldsymbol{\omega}, \mathbf{c}$:

$d_{i,j} = \|(x_i - c_j)\|$ - distance between point and cluster center

σ – scaling parameter ($\sigma > 0$)

$$S_{i,j} = e^{-\frac{d_{i,j}^2}{2\sigma^2}}$$

affinity between the point $\boldsymbol{i}$ and the center $\boldsymbol{j}$. It is large, if they are close in σ units, and small if they are far apart

Soft Assignment (K-Means Clustering)

Now to obtain weights of a particular point, we ensure that sum of cluster weights is equal to 1

$$\omega_{i,j} = \frac{s_{i,j}}{\sum_{l=1}^{k} s_{i,l}}$$

Finally, re-estimated cluster centers could be as:

$$\frac{\sum_i \omega_{i,j} \mathbf{x}_i}{\sum_i \omega_{i,j}}$$

Basically, k-means algorithm is a special case of soft assignment, when $\sigma \to 0$

Soft Assignment (K-Means Clustering)

Algorithm

Choose $\boldsymbol{k}$ initial points $\boldsymbol{c_j}$ and assign it as initial cluster centers and:

Repeat until converge

1. For each pair of data point and cluster compute affinity $\boldsymbol{S_{i,j}} = \boldsymbol{e}^{-\frac{d_{i,j}^2}{2\sigma^2}}$

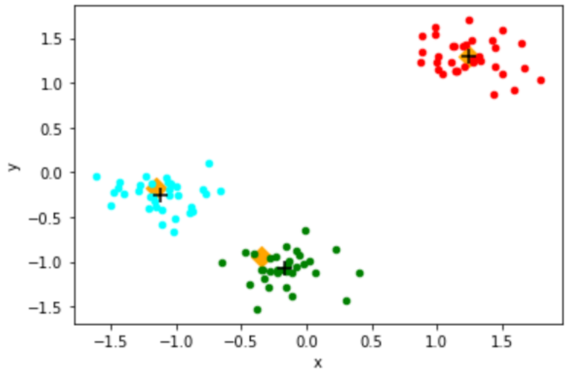
2. Compute soft-weight connections $\boldsymbol{\omega_{i,j}} = \frac{S_{i,j}}{\sum_{l=1}^{k} S_{i,l}}$

3. Get new center for each cluster $\boldsymbol{c_i} = \frac{\sum_i \omega_{i,j} \mathbf{x}_i}{\sum_i \omega_{i,j}}$

Code example

k-means clustering example

kMeansClusteringExample.ipynb

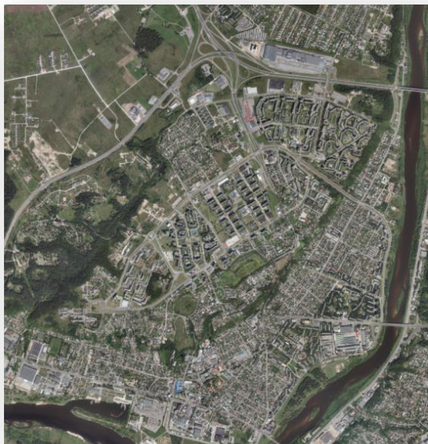


TODO:

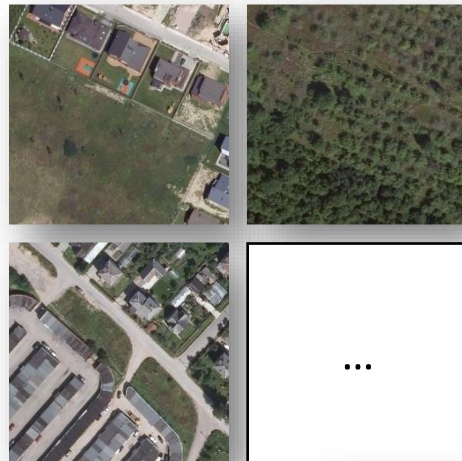
1. use different initial data distributions
2. try random centroid initialization
3. try clustering with real dataset
4. implement selection of number of clusters

Code example

Dataset preparation example for LD2 (image cropping and feature extraction)



1. cropping



2. Features extraction

dataset.txt - Notepad

File Edit Format View Help

filename	b	g	r
croped_1.jpg	2550.0	2357.0	1925.0
croped_10.jpg	1001.0	1097.0	813.0
croped_100.jpg	965.0	893.0	739.0
croped_1000.jpg	685.0	660.0	623.0

LD2_data_preparation.zip

Questions?

ktu
1922